



**UNIVERSIDADE  
ESTADUAL DO  
MARANHÃO**



**Universidade Estadual do Maranhão  
Centro de Ciências e Tecnologia  
Departamento de Computação  
Engenharia de Computação**

**Aline Mariana Barros Silva  
Daniel Pereira Oliveira  
João Victor Alves  
Joerllen Melonio  
José Nathan Araújo**

**Sistema de Gerenciamento de Estoque: em JAVA**

**Documentação**

**São Luís  
2018**

**Aline Mariana Barros Silva**  
**Daniel Pereira Oliveira**  
**João Victor Alves**  
**Joerllen Melonio**  
**José Nathan Araújo**

**Sistema de Gerenciamento de Estoque: em JAVA**

Relatório apresentado ao Prof(a): Dayse,  
como parte dos requisitos necessários à  
obtenção da 3ª nota.

**São Luís**  
**2018**

## **Lista de ilustrações**

Figura 1 – Menu e Cadastro do Produto . . . . .	6
Figura 2 – Trechos do Código . . . . .	6
Figura 3 – Venda do Produto . . . . .	7
Figura 4 – Alterar Cliente e Funcionário . . . . .	7
Figura 5 – Alterar Produto . . . . .	7

## Sumário

<b>1</b>	<b>Sobre o Algoritmo . . . . .</b>	<b>4</b>
<b>2</b>	<b>Adversidades Encontradas . . . . .</b>	<b>5</b>
<b>3</b>	<b>Secões Principais . . . . .</b>	<b>6</b>
<b>3.1</b>	<b>Cadastrar . . . . .</b>	<b>6</b>
<b>3.2</b>	<b>Pesquisar . . . . .</b>	<b>6</b>
<b>3.3</b>	<b>Vender . . . . .</b>	<b>6</b>
<b>3.4</b>	<b>Alterar . . . . .</b>	<b>7</b>
<b>4</b>	<b>Considerações Finais . . . . .</b>	<b>8</b>

## **1 Sobre o Algoritmo**

O algoritmo trata-se de um sistema de vendas de eletrônicos implementado na linguagem de programação JAVA. O objetivo principal desse algoritmo é representar um sistema de funcionamento comercial, no qual são utilizadas funcionalidades como: Cadastro de produtos, Pesquisar produtos, Remover, Vender, Alterar, Exibir, Salvar e Carregar estoque, assim fazendo o uso das ferramentas ministradas em sala de aula, adquiridas ao longo do curso e outras fontes de informação.

## **2 Adversidades Encontradas**

Durante a elaboração do algoritmo, a manipulação de classes e suas interações desafiaram o desempenho e eficiência do algoritmo. Tendo em vista o fato supracitado, foi necessária a realização de um escopo do projeto, para a simplificação e melhor entendimento do funcionamento das operações.

A implementação do algoritmo foi realizada mediante o solucionamento dos itens acima, chegando ao seu resultado final. Desse modo, foram abordados nesse documento os itens mais importantes para realização do algoritmo.

### 3 Secções Principais

#### 3.1 Cadastrar

O primeiro passo do usuário em relação à interação com o sistema é fazer o cadastro de clientes, funcionários e produtos, todos limitados em quantidade. Segue abaixo o menu de interação e os respectivos itens necessários para o cadastro de clientes, funcionários e produtos:

**Figura 1 – Menu e Cadastro do Produto**

```
public static void menu()
{
    System.out.println("=====Menu da Loja=====");
    System.out.println("1 - Novo Cliente");
    System.out.println("2 - Novo Funcionario");
    System.out.println("3 - Pesquisar Cliente");
    System.out.println("4 - Pesquisar Funcionario");
    System.out.println("5 - Alterar dados do Cliente");
    System.out.println("6 - Alterar dados do Funcionario");
    System.out.println("7 - Novo Produto");
    System.out.println("8 - Mostrar Estoque de Produtos");
    System.out.println("9 - Alterar Produto - Só funcionario");
    System.out.println("10 - Pesquisar Produto");
    System.out.println("11 - Vender");
    System.out.println("12 - Consultar Vendas");
    System.out.println("13 - Sair");
    System.out.println("Digite sua opção!");
    System.out.println("=====");

    case 7:
        System.out.println("Digite uma Descrição:");
        String desc = t.next();
        System.out.println("Digite seu Código :");
        long cod = t.nextLong();
        System.out.println("Digite Seu Preço:");
        float preco = t.nextFloat();
        System.out.println("Digite sua Quantidade do Estoque:");
        int qta = t.nextInt();
        prod.addProduto(desc, cod, preco, qta);
        lista_pro.novoProduto(prod);
        break;
}
```

#### 3.2 Pesquisar

Para que se possa pesquisar um cliente, foi criada a função busca. Para isso, foi necessário utilizar o comando *equals*, que verifica se duas variáveis apontam para o mesmo objeto ou se ambos objetos têm valores iguais. É importante lembrar que, ao usar-se o comando em uma classe, deve-se reimplementá-lo.

**Figura 2 – Trechos do Código**

```
case 10 :
    System.out.println("Digite um Produto para Pesquisar:");
    String busca2 = t.next();
    lista_pro.busca(busca2);

    public void busca(String a)
    {
        for(int i = 0 ; i<getQta();i++)
        {
            if(a.equals(this.lista[i].getNome()))
            { System.out.println("Pessoa Encontrada");
              this.lista[i].apresentar();
            }
        }
    }
}
```

#### 3.3 Vender

Para realizar vendas, deve-se considerar que a quantidade do produto no estoque. Sendo assim, tendo este um limite e devendo ser maior que zero.

Considerando a quantidade máxima do produto, caso o produto pedido não esteja no estoque, a venda não poderá ser realizada, sendo o item enviado para uma lista de espera. No menu, local de interação do usuário, foram considerados o tipo do

produto e a quantidade do mesmo. Devendo o produto existir no catálogo da loja, se isto não for válido não é possível prosseguir com a operação.

Figura 3 – Venda do Produto

```
public class Venda {
    private Produto obj;
    private int quant;
    private float preco_total;
}

public Venda(Produto obj, int quant) {
    this.obj = obj;
    this.quant = quant;
    if(obj.getQuantidade() > 0 && obj.getQuantidade() > quant)
    {
        setPreco_total(obj,quant);
        obj.vendeuProduto(quant);
    }
    else
    {System.out.println("Não pode realizar a venda, Produto Insuficiente");
}

case 11:
    System.out.println("Deseja comprar o que ?");
    String dt = t.next();
    int r=lista_pro.posicao(dt);
    if(r == 0)
        System.out.println("Produto não encontrado");
    else
    {
        System.out.println("Quantidade Desejada?");
        int y= t.nextInt();
        Venda vend = new Venda(lista_pro.lista[r],y);
        pp.novoVenda(vend);
    }
}
```

### 3.4 Alterar

Durante o processo de cadastro de clientes, funcionários e produtos, caso o usuário queira alterar informações destes citados, o menu dispõe de opções abrangentes para que cada informação seja alterada. Sendo assim, após a alteração e nova exibição das listas de cadastro, as informações constarão alteradas. Segue abaixo essas operações no menu da loja:

Figura 4 – Alterar Cliente e Funcionário

```
case 4:
    System.out.println("Digite o Funcionario para Alterar:");
    String alit = t.next();
    int p = lista_fun.posicao(alit);
    System.out.println("Deseja Alterar que Atributos: 1 - Nome 2 - Sexo 3 - CPF 4 - Senha 5 - Idade");
    int a = t.nextInt();
    switch(a)
    {
        case 1:
            System.out.println("Digite o novo Nome - ");
            String nome = t.next();
            lista_fun.lista[p].setNome(nome);
            break;
        case 2:
            System.out.println("Digite o novo Sexo - ");
            String sexo = t.next();
            lista_fun.lista[p].setSexo(sexo);
            break;
        case 3:
            System.out.println("Digite o novo CPF - ");
            String cpf = t.next();
            lista_fun.lista[p].setCpf(cpf);
            break;
        case 4:
            System.out.println("Digite a senha Anterior - ");
            long senhaAnt = t.nextLong();
            if(lista_fun.lista[p].getSenha() == senhaAnt)
            {
                case 5:
                    System.out.println("Digite o Cliente para Alterar:");
                    String alit = t.next();
                    int b = lista_cli.posicao(alit);
                    System.out.println("Deseja Alterar que Atributos: 1 - Nome 2 - Sexo 3 - CPF 4 - Senha 5 - Idade");
                    int a = t.nextInt();
                    switch(a)
                    {
                        case 1:
                            System.out.println("Digite o novo Nome - ");
                            String nome = t.next();
                            lista_cli.lista[b].setNome(nome);
                            break;
                        case 2:
                            System.out.println("Digite o novo Sexo - ");
                            String sexo = t.next();
                            lista_cli.lista[b].setSexo(sexo);
                            break;
                        case 3:
                            System.out.println("Digite o novo CPF - ");
                            String cpf = t.next();
                            lista_cli.lista[b].setCpf(cpf);
                            break;
                        case 4:
                            System.out.println("Digite a senha Anterior - ");
                            long senhaAnt = t.nextLong();
                            if(lista_cli.lista[b].getSenha() == senhaAnt)
                            {
                                System.out.println("=====Funcionario Logado=====");
                                System.out.println("Digite oCodigo do Produto para Alterar:");
                                long codg = t.nextLong();
                                int pl = lista_pro.posicao(codg);

                                System.out.println("Deseja Alterar que Atributos: 1 - Preço 2 -Quantidade no Estoque");

                                int esq = t.nextInt();
                                switch(esq)
                                {
                                    case 1:
                                        System.out.println("Digite o novo Preço! - ");
                                        float preco1 = t.nextFloat();
                                        lista_pro.lista[pl].setPreco_unitario(preco1);
                                        break;
                                    case 2:
                                        System.out.println("Digite o nova Quantidade! - ");
                                        int qtc = t.nextInt();
                                        lista_pro.lista[pl].setQuantidade(qtc);
                                        break;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Figura 5 – Alterar Produto



#### **4 Considerações Finais**

Portanto, foram abordados as adversidades encontradas para a elaboração do algoritmo e a solução para as mesmas, destacando-se os itens mais importantes para a manipulação do usuário. Visando a simplicidade de manuseio e praticidade, o projeto visou a acessibilidade e descomplicação de operações que devem ser realizadas em um sistema de vendas.